Human Factors in Design, Engineering, and Computing, Vol. 159, 2024, 711–720
https://doi.org/10.54941/ahfe1005638

Autonomous Behavior of Bipedal Robot by Learning Time-Series Camera Images

Manabu Motegi

Takushoku University, 815–1 Tatemachi, Hachioji-shi, Tokyo 193-0985, Japan

ABSTRACT

The author is conducting basic research on the autonomous behavior of a small biped robot. The system under study acquires behavioral data when a human controls the small biped robot. This system then learns from this behavioral data and image data obtained from the robot's onboard camera. However, our previous method did not account for time-series behaviors, resulting in the repetition of certain behaviors. To address this issue, this paper utilizes Recurrent Neural Network (RNN), which are well-suited for learning time-series information. As a result, it was confirmed that the robot could behave autonomously without frequently repeating specific behavioral patterns.

Keywords: Machine learning, Camera image, Autonomous behavior

INTRODUCTION

We constructed a system that combines a Convolutional Neural Network (CNN), a Recurrent Neural Network (RNN), and a Support Vector Machine (SVM) using camera images as input and confirmed the autonomous behavior of a biped robot.

Since late 2019, COVID-19 has spread rapidly, restricting activities for people worldwide. To prevent infection, it became necessary to find ways for people to continue their daily activities without physical contact. However, certain challenges cannot be fully addressed by teleconferencing systems, particularly in fields that require direct interaction with the physical world, such as nursing care, logistics, travel, and daily activities. It is difficult to use robots that do not have autonomy and only have remote control functions. This is because of the heavy burden on the operator who constantly remotely controls the robot. Currently, robots can't behave autonomously in any environment without human remote control using existing technology. Therefore, the author studied a learning structure that enables robots to behave autonomously by learning from the robot operation logs of human operators (Motegi, 2023). However, this method did not consider the time series of behaviors. As a result, some behavior patterns were repeated during autonomous behaviors, such as right turn, left turn, right turn, and so on. Hence, in this paper, we aimed to address the time series of behaviors and studied a learning structure to suppress the repetition of these behavior patterns during the autonomous behavior of the robot.

PREVIOUS WORK

Research on autonomous behavior for robots, drones, and automobiles has been actively conducted in recent years. However, many of these methods rely on various sensors in addition to camera images, which can be costly. Also, the algorithms for autonomous behavior tend to be complicated (Nieuwenhuisen, 2014). Traditional image-based navigation requires a multi-step process. For example, the first step involves extracting features from a camera image (Vale, 2004). Next, a map is created using these results (Jeong, 2006). Finally, the action is determined based on the rules that were established in advance (Belker, 2002). The above is the general method (Kim, 2018). However, in these multi-step processes, adjustments are needed at each stage when the environment changes. Additionally, errors can occur at each stage and accumulate over time. Research has also been conducted to address these issues using deep learning, which takes a camera image as input and directly generates the output (Kim, 2018) (Liu, 2017). This is known as the end-to-end method.

Hence, the author conducted basic research on learning from the human operation logs so that the robot can behave autonomously (Motegi, 2023). However, this method showed a phenomenon in which several patterns of behavior were repeated during autonomous behavior. Therefore, in this paper, we further investigate the machine learning structure with the aim of suppressing this phenomenon.

SYSTEM REQUIREMENTS

As mentioned earlier, both camera images and human operation logs are used to enable the robot to behave autonomously. Therefore, the system must understand when and how the robot is being operated by humans. Furthermore, the robot should avoid unnecessarily repeating certain behavior patterns such as right turn, left turn, right turn, and so on, as much as possible. In addition, it is desirable for the robot to behave without colliding with the environment during its autonomous behavior. Therefore, we have outlined the requirements for this system as follows.

(1) It is possible to acquire both camera images and human operation logs (logs acquisition).

(2) To prevent the repetition of meaningless behavior patterns, the time series of selected behaviors operated by the human should be able to be reflected during autonomous behavior (reflection of time-series behaviors).

In this paper, we specifically consider the above requirement (2). Then, the following requirements were evaluated.

(3) Autonomous behavior without colliding with the environment is possible (autonomous behavior determination).

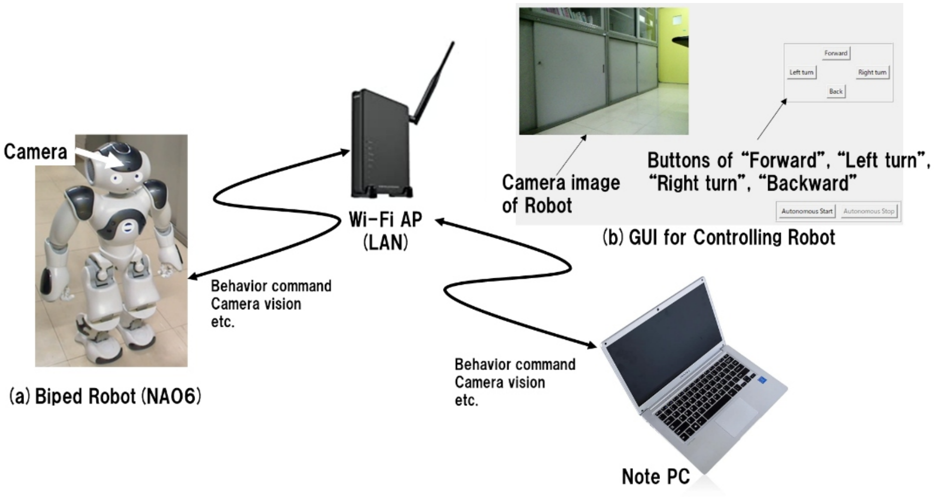


Figure 1: System architecture: Robot (NAO6) and GUI used in the experiment.

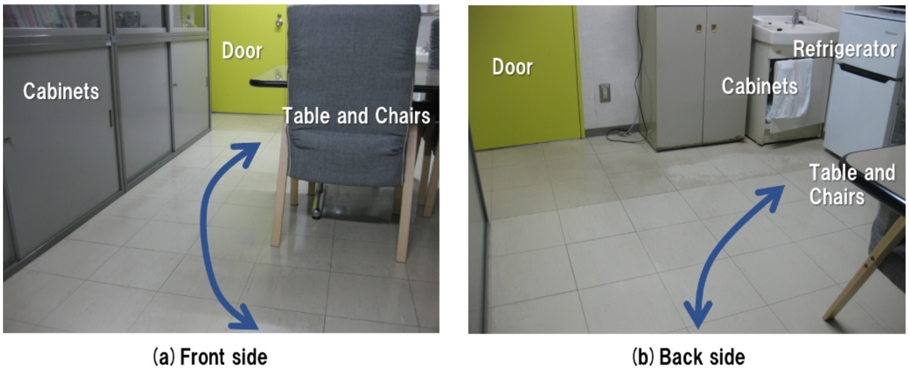


Figure 2: Experimental environment.

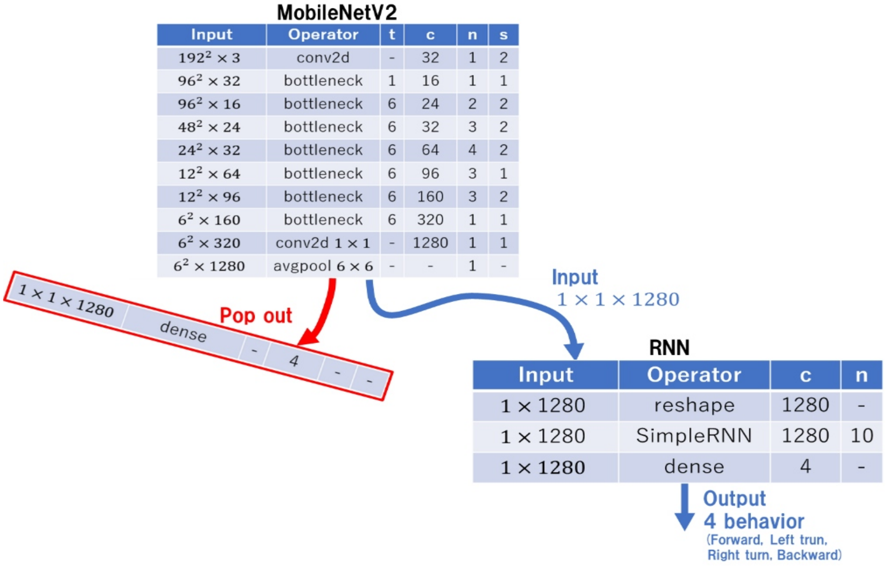
SYSTEM IMPLEMENTATION AND EXPERIMENT

Figure 1 presents the system architecture, the robot utilized in the experiment, and the graphical user interface (GUI) for operating the robot. Figure 2 shows the experimental environment.

Figure 3 illustrates the structure of the learning component, which integrates SVM into the convolutional neural network. Figure 4 depicts the structure of the convolutional neural network with RNN added. Figure 5 shows an example of the accuracy values when training with the structure shown in Figure 4. Figure 6 shows the structure of Figure 4 with additional SVM. Table 1 shows an example of the confusion matrix of SVM trained with the structure shown in Figure 6.

Implementation for Requirements (1) (Logs Acquisition)

First, we describe the component of the constructed system related to log acquisition, as outlined in requirement (1). Figure 1(a) shows that the robot used in the experiment is the NAO6, manufactured by SoftBank Robotics. The robot was pre-set with primitive behaviors: forward, right turn, left turn, and backward. The parameters of each motion were set to move forward and backward 10 cm each, and to make a right turn and a left turn 10 degrees each.



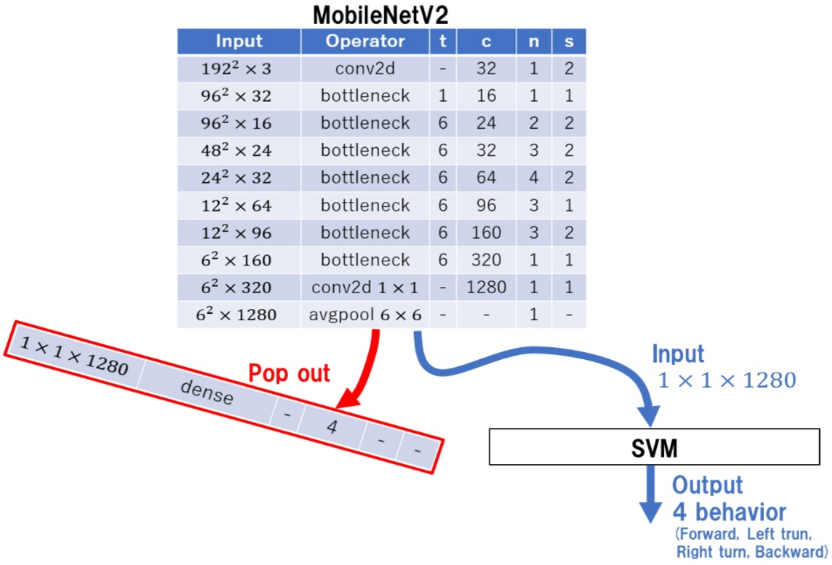
MobileNetV2

Input	Operator	t	c	n	s
$192^2 \times 3$	conv2d	-	32	1	2
$96^2 \times 32$	bottleneck	1	16	1	1
$96^2 \times 16$	bottleneck	6	24	2	2
$48^2 \times 24$	bottleneck	6	32	3	2
$24^2 \times 32$	bottleneck	6	64	4	2
$12^2 \times 64$	bottleneck	6	96	3	1
$12^2 \times 96$	bottleneck	6	160	3	2
$6^2 \times 160$	bottleneck	6	320	1	1
$6^2 \times 320$	conv2d 1 × 1	-	1280	1	1
$6^2 \times 1280$	avgpool 6 × 6	-	-	1	-

RNN

Input	Operator	c	n
1 × 1280	reshape	1280	-
1 × 1280	SimpleRNN	1280	10
1 × 1280	dense	4	-

Figure 3: Structure of the training part with SVM added to CNN.



MobileNetV2

Input	Operator	t	c	n	s
$192^2 \times 3$	conv2d	-	32	1	2
$96^2 \times 32$	bottleneck	1	16	1	1
$96^2 \times 16$	bottleneck	6	24	2	2
$48^2 \times 24$	bottleneck	6	32	3	2
$24^2 \times 32$	bottleneck	6	64	4	2
$12^2 \times 64$	bottleneck	6	96	3	1
$12^2 \times 96$	bottleneck	6	160	3	2
$6^2 \times 160$	bottleneck	6	320	1	1
$6^2 \times 320$	conv2d 1 × 1	-	1280	1	1
$6^2 \times 1280$	avgpool 6 × 6	-	-	1	-

Figure 4: Structure of the training part with RNN added to CNN.

The NAO6 robot is equipped with two cameras, one on its forehead and one on its mouth; however, in this experiment, only the forehead camera was utilized. A notebook PC (FUJITSU LIFEBOOK WA3/D3, with an Intel

Core i7-9750H CPU and 8GB of memory) was used to send operation commands to the NAO6 and for learning purposes. The notebook PC and the NAO6 communicate via a Wi-Fi access point. Additionally, the system was developed using Python as the programming language.

The GUI shown in Figure 1(b) was developed for the experiment. Initially, a human uses this GUI to control the robot. In this system, when the human clicks a behavior button, the selected behavior and the robot's camera image prior to the action are recorded. The experimental environment depicted in Figure 2 is the laboratory at the author's university. The robot was operated using the GUI in this environment, and the selected behavior and images were saved as learning data.

Implementation for Requirements(2) (Reflection of Time-Series Behaviors)

In this section, we first discuss the authors' previous work. In our previous study, we operated the robot three round trips from the front in Figure 2(a) to the refrigerator in Figure 2(b) and obtained a total of 624 images(22 backward, 310 forward, 77 left turns, and 215 right turns).

For the learning data mentioned above, the input was the camera image of the robot, and the output was the behavior selected by the human operator. Fine-tuning was then performed on the convolutional neural network, MobileNet V2 (Sandler, 2018), which had been pre-trained on ImageNet (Deng, 2009). As shown in

Figure 3, after fine-tuning, the fully connected layer responsible for behavior selection was removed from MobileNet V2, and the output of the average pooling layer was used as input to the SVM.

However, this method did not consider the time series of behaviors, and some behavior patterns, such as right turn, left turn, right turn, and so on, were repeated during autonomous behaviors (Motegi, 2023).

To consider the time series of behavior, we decided to utilize RNN. Specifically, as shown in Figure 4, we used RNN instead of SVM in Figure 3 for training. In this case, MobileNetV2 is already fine-tuned with the above training data and is not re-trained, and only the RNN part is trained. Because some of the above training data were not acquired in chronological order, we prepared 659 images (42 backward, 359 forward, 82 left turns, and 176 right turns) for RNN training, which consisted of time-series data from five new round trips in the environment shown in Figure 2. For the RNN implementation, we used SimpleRNN provided by Keras (Google, 2015), a python deep learning library. The 1 × 1 × 1280 data, which is the output of the average pooling layer of MobileNetV2, was formatted to 1 × 1280 and input to the RNN in Figure 4. The number of intermediate layers and the number of samplings of past data for estimating the next behavior were adjusted in various ways. In practice, the number of intermediate layers of the RNN was varied between 80 and 1280, the number of samplings of the considered historical data between 1 and 16, and the number of epochs was trained with 100 or 30. However, the accuracy value for the evaluation data was not improved, being only 0.5 at best. For example, as shown in Figure 5, the final accuracy value for the evaluation data(orange line) was about 0.51 when the number of intermediate layers of RNN was set to 1280, the number of samplings of past data to be considered was 10, and the number of epochs was 30. The blue line in Figure 5 shows the change in accuracy values for the training data. We observed the accuracy values by changing the training data, the number of intermediate layers, and other parameters in the structure shown in Figure 4, but it was difficult to find a good training condition.

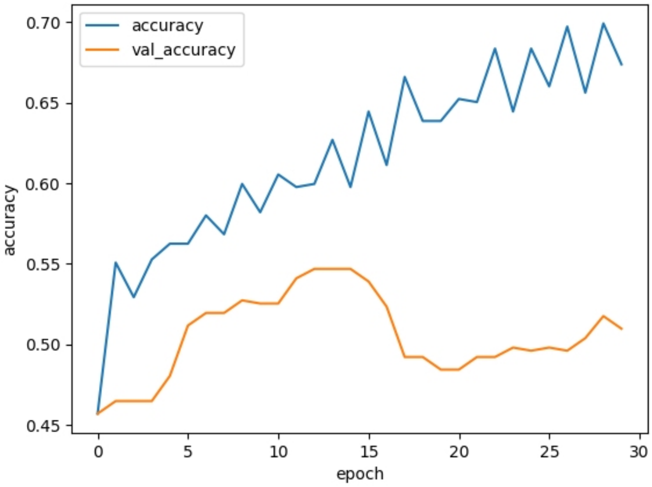
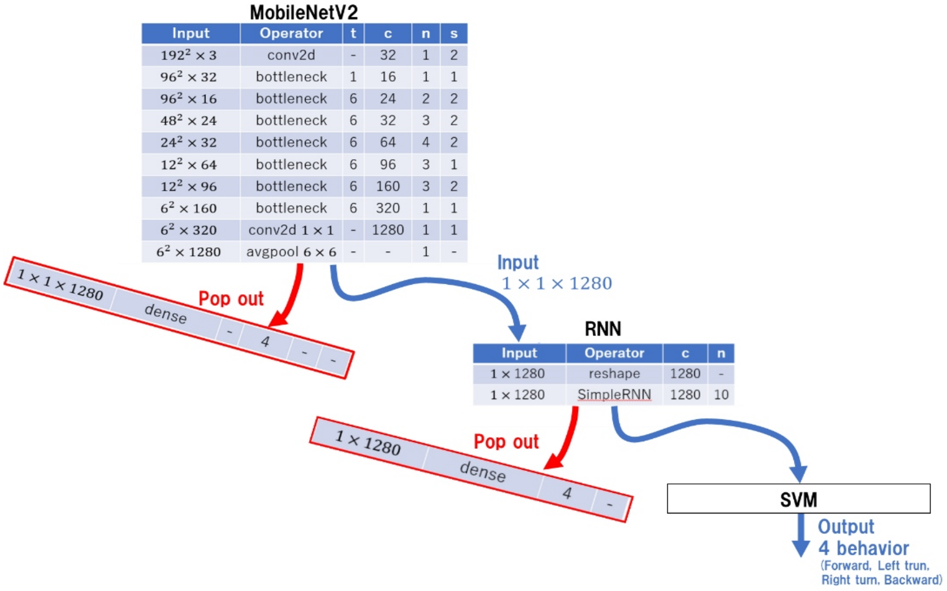


Figure 5: Training results when RNN is added to CNN (1280 intermediate layers, 10 samplings of past data to be considered).



Input	Operator	t	c	n	s
$192^2 \times 3$	conv2d	-	32	1	2
$96^2 \times 32$	bottleneck	1	16	1	1
$96^2 \times 16$	bottleneck	6	24	2	2
$48^2 \times 24$	bottleneck	6	32	3	2
$24^2 \times 32$	bottleneck	6	64	4	2
$12^2 \times 64$	bottleneck	6	96	3	1
$12^2 \times 96$	bottleneck	6	160	3	2
$6^2 \times 160$	bottleneck	6	320	1	1
$6^2 \times 320$	conv2d 1 × 1	-	1280	1	1
$6^2 \times 1280$	avgpool 6 × 6	-	-	1	-

Input	Operator	c	n
1 × 1280	reshape	1280	-
1 × 1280	SimpleRNN	1280	10

Figure 6: Structure of the training part that adds SVM to CNN and RNN.

Table 1. SVM's confusion matrix on the evaluation data.

		Predicted Behaviors			
		Backward	Forward	Left Turn	Right Turn
Correct behaviors	Backward	76%	17%	8%	0%
	Forward	2%	88%	4%	6%
	Left turn	0%	28%	72%	0%
	Right turn	12%	18%	0%	70%

Therefore, as shown in Figure 6, we attempted to determine autonomous behavior by SVM by removing all coupling layers of the RNN. As in the previous example, the number of intermediate layers in the RNN is 1280, and the number of samplings of past data to estimate the next behavior is 10. This was determined empirically because the maximum accuracy value was obtained for the evaluation data when SVM was added to the RNN trained with these parameters adjusted as described above. The output 1×1280 of the RNN was used as input to SVM to perform 4 classifications corresponding to each behavior. The scikit-learn python library was used to use SVM as described above, and the GridSearchCV function was used to determine the SVM parameters.

Table 1 presents the SVM confusion matrix for the evaluation data. The rows in Table 1 represent the true classes, while the columns represent the predicted results. The prediction accuracy for backward behavior was approximately 76%, with 17% of cases incorrectly classified as forward. This misclassification occurred because the experimental environment required minimal backward movement, leading to infrequent use of this behavior by the operator.

Conversely, the prediction accuracy for forward behavior was high, at 88%. This is due to the abundance of forward movement data, with 359 instances recorded during robot operation.

For the left turn, the prediction accuracy was 72%, with 28% of cases mistakenly identified as forward. This may be because, in similar camera images in the training data, there were cases where the operator selected a left turn even when the operator could have selected forward.

For the right turn, 70% of the predictions were correct, 18% were incorrectly identified as forward, and 12% were incorrectly identified as backward. This is because the experimental environment has many obstacles on the left side, leading the operator to frequently choose the right turn in situations where the robot could have moved forward.

SYSTEM EVALUATION

Evaluation of Requirement (3) (Autonomous Behavior Determination)

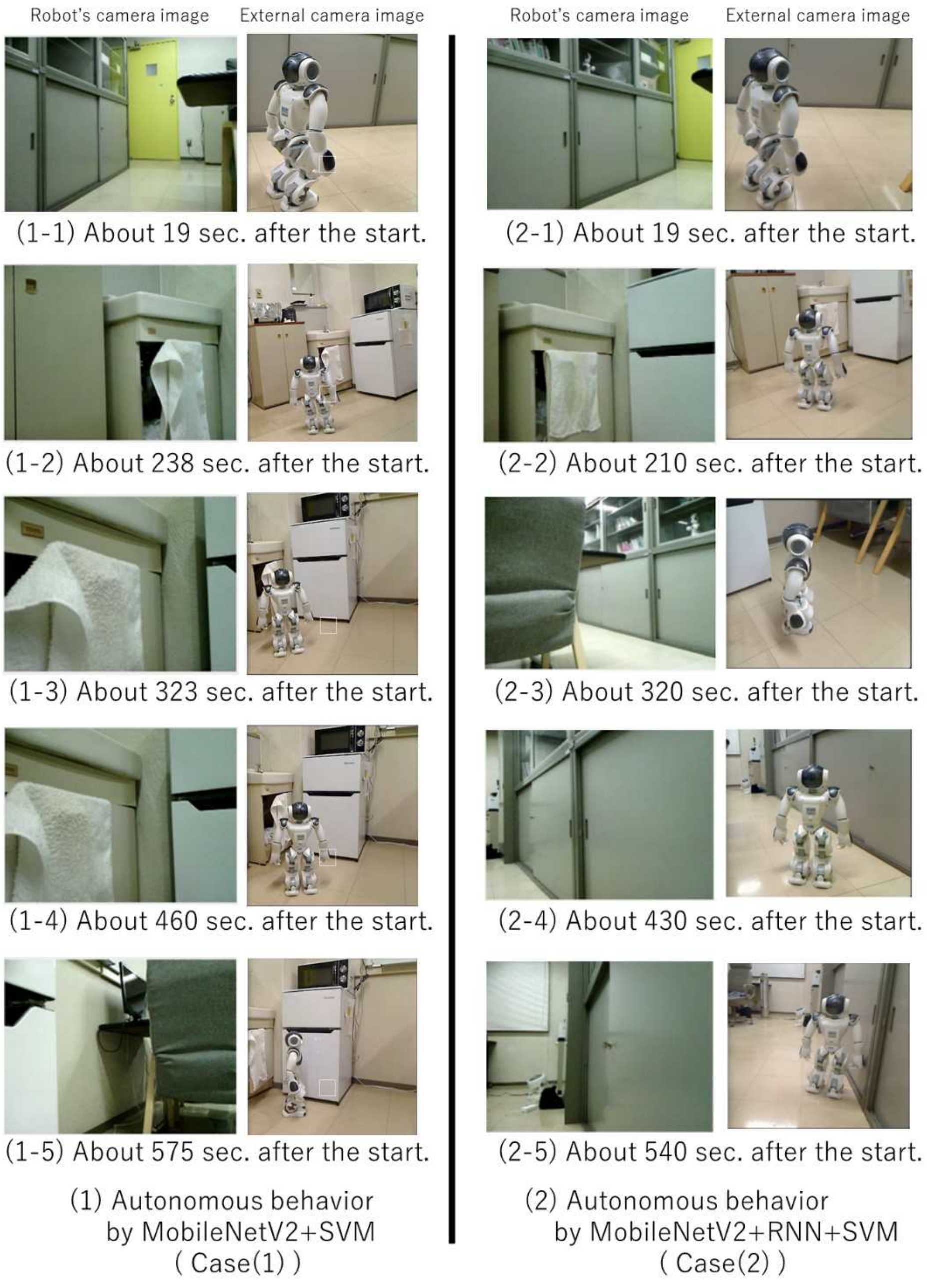
Figure 7 shows a photograph of an autonomous behavior experiment. Concerning the aforementioned requirement(3), the robot was made to behave autonomously using the trained system. For comparison, the robot's position before colliding with the environment was checked in each of the following cases.

(1) The case where the final layer of MobileNetV2 with fine tuning after the 13th block is deleted and its output is classified by SVM as shown in Figure 3 (Motegi, 2023).
(2) The case where the SVM in (1) above is removed and an RNN is added instead, and the output is classified by the SVM as shown in Figure 6.

Figure 7 shows the images of the robot-mounted camera (left column) and the external camera (right column) taken for recording purposes when the robot behaved autonomously according to (1) and (2) above. In both cases, the robot began autonomous behavior from approximately the same position in front of the location shown in Figure 2(a).

As mentioned above, in the above case (1), the robot was trained on data from 3 round trips made by a human between the front in Figure 2(a) and the refrigerator in Figure 2(b). In case (2), in addition to case (1), the RNN and SVM were trained on the data of 5 round trips. Thus, after reaching the refrigerator, the robot is expected to return to the point where it began its autonomous behavior.

However, in case (1) above, the robot behaved autonomously to the refrigerator, but its left arm collided with the refrigerator when it made a right turn (Figure 7(1–5)). Therefore, the experiment was terminated approximately 9 minutes and 30 seconds after the start of autonomous behavior. At that time, as shown in Figure 7(1–3) and Figure 7(1–4), the robot repeatedly performed right-turn and left-turn behaviors near the cabinet where the towel was hung, close to the refrigerator. This occurred because the robot selects primitive behaviors reflexively based only on the onboard camera images, without considering the temporal context of the behaviors. On the other hand, in case (2) above, the right arm collided with the right-side cabinet on the way back to the starting point of autonomous behavior (Figure 7(2–5)) after making a right turn in front of the refrigerator (Figure 7(2–3)). Therefore, the experiment was terminated approximately 9 minutes after the start of autonomous behavior. This is because the system misidentified a left turn as a forward behavior. In this case, the right and left turns were not repeated as in the case (1), but the forward and backward behaviors were repeated several times. However, compared to the above case (1), it did not collide and behaved a long distance in a short time without repeating left and right turns. This confirms the effect of the time-series structure shown in Figure 6.



(1-1) About 19 sec. after the start.

(1-2) About 238 sec. after the start.

(1-3) About 323 sec. after the start.

(1-4) About 460 sec. after the start.

(1-5) About 575 sec. after the start.

(1) Autonomous behavior by MobileNetV2+SVM (Case(1))

(2-1) About 19 sec. after the start.

(2-2) About 210 sec. after the start.

(2-3) About 320 sec. after the start.

(2-4) About 430 sec. after the start.

(2-5) About 540 sec. after the start.

(2) Autonomous behavior by MobileNetV2+RNN+SVM (Case(2))

Figure 7: Picture of autonomous behaving.

CONCLUSIONS

As a fundamental study toward realizing a real-world avatar, we investigated a system that selects autonomous behavior by learning from camera images and human operation logs. The system combines a fine-tuned MobileNetV2, a recurrent neural network (RNN), and a Support Vector Machine (SVM).

The autonomous behavior of the robot up to the point of collision with the environment was compared between this and a system that did not include an RNN. When the latter RNN is not included, the time series of the behavior is notconsidered. Therefore, right and left turns were repeated during autonomous behavior. However, as proposed in this study, combining this with an RNN resulted in a learning mechanism that considers the time series of behaviors. This enabled suppression of the phenomenon of repeated left-right turns during autonomous behavior. It was also confirmed that the proposed system can behave autonomously over long distances without colliding with the environment.

REFERENCES

A. Vale, J. M. R., 2004. Feature extraction and selection for mobile robot navigation in unstructured environments. *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles,* 37(8), pp. 102–107.

C. Liu, B. Z. C. W. Y. Z. A. F. H. L., 2017. CNN-Based Vision Model for Obstacle Avoidance of Mobile Robot. *MATEC Web of Conferences.*

Google, 2015. *Keras Documentation.* [Online] Available at: https://keras.io[Accessed 7 6 2024].

J. Deng, W. D. R. S. L.-J. L. K. L. a. L. F.-F., 2009. ImageNet: A Large-Scale Hierarchical Image Database. *IEEE Computer Vision and Pattern Recognition (CVPR),* pp. 2–9.

M. Sandler, A. H. M. Z. A. Z. L.-C. C., 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 4510–4520.

M. Motegi, 2023. Autonomous Behavior of Biped Robot by Learning Camera Images. *HCI International 2023 Posters,* pp. 498–506.

M. Nieuwenhuisen, D. M. S. B., 2014. Obstacle Detection and Navigation Planning for Autonomous Micro Aerial Vehicles. *International Conference on Unmanned Aircraft Systems,* pp. 1040–1047.

T. Belker, D. S., 2002. Local Action Planning for Mobile Robot Collision Avoidance. *Proceedings of the IEEE/RSJ Inti. Conference on Intelligent Robots and Systems,* pp. 601–606.

W. Y. Jeong, K. L., 2006. Visual SLAM with Line and Corner Features. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2570–2575.

Y-H. Kim, J.-I. J. S., 2018. End-to-End Deep Learning for Autonomous Navigation of Mobile Robot. *IEEE International Conference on Consumer Electronics.*